#16
6/5/03
A.J.

GLENN | PATENT | GROUP

FACSIMILE TRANSMITTAL SHEET

TO:	FROM:
Hal Dodds	Phone: (650) 474-8400 Fax: (650) 474-8401
COMPANY:	DATE:
PTO	5/30/2003
FAX NUMBER:	TOTAL NO. OF PAGES INCLUDING COVER:
703-746-7239	3
PHONE NUMBER:	SENDER'S REFERENCE NUMBER:
703-305-1802	NETS0059
RE: 09/447,443	YOUR REFERENCE NUMBER:

Official

☒ URGENT ☒ FOR REVIEW ☐ PLEASE COMMENT ☐ PLEASE REPLY ☐ PLEASE RECYCLE

Hal – attached is Exhibit A "An Elegant Method to Enable User Manageability in a LDAP Server" as referenced is the Declaration of Prasanta Behera.

Please contact me if you have any questions.

Thank you,

Rhonda Dunn

Rhonda Dunn

3475 EDISON WAY, STE. L, MENLO PARK, CA 94025 TEL: (650) 474-8400 FAX: (650) 474-8401

EXHIBIT A

An Elegant Method to Enable User Manageability in a LDAP Server

Prasanta Behera
September 2, 1998

Background
An LDAP directory (such as Netscape's Directory Server) is a collection of *Entries*. Each entry has a name (called the Distinguished Name) and a list of attribute values. In a *directory* the entries are organized in a tree structure, with major groupings that are subdivided into smaller units. A directory might contain several *organization* entries each of which contains several *organizationalUnit* entries. These may then be further subdivided. LDAP provides search operations that can be performed over specified portions of the directory tree. Therefore trees and subtrees are a natural way to deal with data stored in the LDAP directory.

Entries and attributes correspond to a wide variety of data types, such as personnel information, server configuration, business relationships and user preferences, among others. Since all the entries are stored within a single directory, a method is required to restrict the availability of specific information to authorized users. The method used to control acces is via Access Control Lists (ACL). The Directory Server Administrator (DSAdmin) will create some basic ACL rules which grant permission to various people to various information in the directory. Most of these security considcration will require from a few tens to couple of hundred rules. The smaller number of ACL rules the better is the performance and easier to manage.

Since directory is a critical central repository in an intranet containing information about people (from now we will focus on people as it is easy to explain), it is imperative that a rich set of options/features be provided. For example, the user should be able to modify his/her entry. For example, allow the user the ability to update home address or home phone number without the DSAdmin intervention. A better thing will be the ability to let the user decide who can access some of his/her personnel information. The only way to do that is to allow the users to create ACLs. Now a directory can contain millions of entries like the one Netcenter is using. To support what we want will require millions of ACLs and which will not only make the server performance bad but also will be highly unmanageable. It also provide a risk i.e. the user can create a rule denying the DSAdmin some privilege which is a bad. Another disadvantage is that user may have to learn the ACL syntax which can be very complex.

So, the problems we have here is:

- How to let the users manage some of their own information
- How the DSAdmin can manage the information so that no security rules are violated.
- How not to make the server manageable under the above circumstances.

In a realistic sense, a DSAdmin will like to have the following rules. Out of n attributes in the directory

1. Allow n1 attributes to be read by anyone in the world (that's a typical need Example attributes are cn, sn, phonenumber).
2. Allow n2 attributes to be read and modifiable by the self (Ex: home address, home phonenumber)
3. Allow n3 attributes to be managed by a owner/manager (ex: salary, employeegrade)
4. Allow n4 attributes to be managed by the user i.e. the user decides who can read or modify the attributes (ex: The user can decide only Sam can read his hobbies. Only Kelly can read or change emergency contact info so that she can help in case of need. <may need better example>)
5. Do not allow the rest n5= [n - (n1 + n2 + n3 + n4)] attributes to the general public except the Administrator group (ex: employee status).

Solving 1, 2, 3 & 5 are pretty easy. The Netscape Directory server has implemented new features in which a single ACL can achieve each of the items. The only remaining item is the 4th case. As we said, to do that you will need to provide the ability to the user to create it's own ACLs. This could lead to millions of ACLs which is not acceptable. This document describes a novel approach to achieve that with a few ACL and use an existing internet standard to achieve that. The advantages would be obvious by now.

One question which arises is where this is applicable. We have a need right now for this. The Netscape netcenter has a registry of netcenter members. All those member information is stored in a directory server. The netcenter not only would like to keep the member's information/profiles but also provide flexibility to let members keep some other key information which other members can access. There is a big need now and no solution exits which can take care of it in an elegant way. Any directory application can take advantage of this.

The Solution:

A directory entry will closely resemble like this
dn: uid=prasanta,ou=People,o=Netscape Communications Corp.,c=US
sn: Behera
uid: prasanta
mail: prasanta@netscape.com
telephonenumber: 9999
objectclass: top

objectclass:person
cn: Prasanta Behera
postaladdress: 501 E. Middlefield Road, Mountain View, CA 94043, USA
givenname: Prasanta
manager: uid=claire, ou=People,o=Netscape Communications Corp.,c=US

The LDAP standard is very flexible. It allows to extend the schema by adding new attributes or objectclass. So, one can add a new attribute called "mypersonelInfo" to the entry as long as the objectclass which has that attribute is added in.

We will use the following ACL syntax to explain how the cases 1, 2, 3, & 5 can be achieved now propose the solution for the 4th case. Note, the syntax is used for reference only.

- Allow n1 attributes to be read by anyone

  ACL: (list of n1 attrs) (allow (read) user = "anyone")

- Allow n2 attributes to be read/writable by self

  ACL: (list of n2 attrs) (allow (read, write) user = "self")

- Allow n5 attrs to be read/writable by by the admin group only

  ACL: (list of n5 attrs) (allow (read, write) group = "Admingroup")

- Allow the owner or the manager to manage the n3 attributes

  ACL:(list of n3 attributes) (allow (read, write) attr="manager" or attr = "owner")

  This needs a little explanation. For the above entry (i.e. my entry), my manager is Claire and she can read/write those n4 attributes. Similarly Joe' manager Bill can read/modify Joe's n4 attributes. This all can be achieved by one ACL. The value of the "manager" is plugged in at runtime.

Out of the n4 attributes a finer granularity can be achieved i.e., some of the n4 attributes can be only read by someone (n4-read attrs) and some can be modifiable by someone (n4- write attrs). The solution to achieve the 4th case is to create an acl like (using the example discussed earlier)

ACL: (list of n4-read attrs) (allow (read) filterattr= "readusercontrolattr")
Ex: (hobby, emergencyContact) (allow (read) filterattr= "readusercontrolattr")

ACL: (list of n4-write attrs) (allow (write) filterattr= "writeusercontrolattr")
Ex: (emergencyContact) (allow (write) filterattr= "writeusercontrolattr")

where the value of a readusercontrolattr & "writeusercontrolattr" will look like
readusercontrolattr: (|(uid=sam) (uid=kelly))
writeusercontrolattr:(uid=kelly)

The value of these attributes are a LDAP Filter which an internet standard (RFC 2254). The ACLs are created by the DSAdmin. This way s/he has full control of what information the user can give out. In the server, the value of "writeusercontrolattr" is plugged at the runtime with "(uid=kelly)". So, if Kelly is the client, the filter matches to TRUE and Kelly is allowed to modify the "emergencyContact" attribute. However, if Bill is the client, the filter matches to FALSE and Bill is denied the privilege. Each user can now create LDAP Filters which will allow them to manage their own information.

Advantages:

- The Admin has complete control of what a user can do.
- Only a handful of ACLs needed instead of Millions.
- The performance of the server will be better.
- The value of the new attribute are based on internet standard.